ALGORITMOS Y ESTRUCTURAS DE DATOS III - 1$^{\text{er}}$ Parcial
Fecha examen: 09-MAY-2018 / Fecha notas: 23-MAY-2018

	Nº Orden	Apellido y nombre	L.U.	Cant. hojas[1]
Completar:	274	Brandwein Eric	349/16	7
	Nota (Nº)	Nota (Letras)	Docente	
No completar:	9	NUEVE	ARIEL	

1. Sea G un grafo de m ejes. Demostrar que para todo $d \in \mathbb{N}$, si $m < d \times (d+1)/2$ entonces existe un vértice v tal que $d(v) < d$. 2 p.

2. Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, se define su grafo junta como $G_1 + G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$, es decir, el grafo que contiene a G_1 y a G_2 como subgrafos, y además contiene un eje entre cada vértice de G_1 y cada vértice de G_2.

   Determinar para qué valores de n, p, q y h los siguientes grafos son grafos junta. Justificar. c/u 0.4 p.

   (a) K_n
   (b) C_n (ciclo simple de $n \geq 3$ vértices)
   (c) P_n (camino simple de n vértices)
   (d) $K_{p,q}$
   (e) árbol binario completo de altura $h \geq 0$

3. Sea $G = (V, E)$ un digrafo de n vértices y m ejes. Existen varias estructuras de datos que permiten representar a G. Si se utilizan *listas de sucesores*, para cada vértice $v \in V$ se dispone de una lista donde aparecen todos los vértices $w \in V$ tales que $(v, w) \in E$. De manera similar, en las *listas de predecesores*, para cada vértice $v \in V$ se tiene una lista donde aparecen todos los vértices $w \in V$ tales que $(w, v) \in E$. 2 p.

   Diseñar un algoritmo para cada uno de los problemas indicados, que tenga la complejidad mencionada en cada caso. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad. Justificar.

   (a) Dado G representado con listas de sucesores, representarlo con listas de predecesores, con complejidad $O(m+n)$.
   (b) Dado G representado con listas de sucesores, invertir todos los ejes y representar el resultado con listas de sucesores, con complejidad $O(m+n)$.
   (c) Dado G representado con listas de sucesores, invertir todos los ejes y representar el resultado con listas de predecesores, con complejidad estrictamente mejor que $\Theta(m+n)$.

4. (a) Sea G un grafo conexo con pesos no negativos asociados a sus ejes. Sea T un árbol generador mínimo de G, y sea $e = (u, v)$ un eje de T. Demostrar que e es un camino mínimo en G entre u y v. 1.5 p.
   (b) ¿Sigue valiendo la propiedad del punto anterior si los pesos pueden ser negativos? En caso afirmativo demostrar; en caso negativo dar un contraejemplo y justificar. 0.5 p.

5. Viajante de comercio bitónico: Se tienen $n \geq 2$ puntos en el plano $p_1, p_2, \ldots p_n$, ordenados de manera creciente de acuerdo a su coordenada x (no hay dos puntos con la misma coordenada x). Un recorrido bitónico de los puntos es un recorrido que comienza en p_1, recorre de manera creciente en x algunos de los puntos hasta llegar a p_n (ida), y finalmente recorre de manera decreciente en x algunos de los puntos hasta volver a p_1 (vuelta); el recorrido pasa exactamente una vez por cada punto. 2 p.

   Para $n = 2$ hay un único recorrido bitónico, que es p_1, p_2, p_1. Para $n = 3$ hay dos recorridos bitónicos. Uno es p_1, p_2, p_3, p_1, mientras que el otro es p_1, p_3, p_2, p_1. Sin embargo, ambos recorridos tienen la misma longitud, ya que lo que recorre uno a la ida, lo recorre el otro a la vuelta, y viceversa. En general, dado cualquier recorrido bitónico, existe uno "simétrico" con la ida y la vuelta intercambiadas.

   Diseñar un algoritmo que encuentre la mínima longitud que puede tener un recorrido bitónico. El algoritmo debe tener complejidad temporal y espacial $O(n^2)$ y estar basado en programación dinámica. Mostrar que el algoritmo propuesto es correcto y determinar su complejidad (temporal y espacial). Justificar. El mejor algoritmo que conocemos tiene complejidad temporal $O(n^2)$ y espacial $O(n)$.

   SUGERENCIA: Para $i \neq j$, sea $f(i,j) = f(j,i)$ la longitud mínima que puede tener un camino que vuelve de p_i a p_1 y va de p_1 a p_j, pasando exactamente una vez por cada punto $p_1, p_2, \ldots p_{\max(i,j)}$. Basta sumar a $f(i,n)$ la distancia entre p_n y p_i para tener la longitud del recorrido bitónico $p_1, \ldots p_n, p_i, \ldots p_1$.

[1] Incluyendo a esta hoja. Entregar esta hoja junto al examen.